Fast & Accurate Calorimeter Simulation With Diffusion Models

Oz Amram & Kevin Pedro
The Need for Fast Simulation

- Geant4 calo simulation is a significant part of ATLAS computing budget
  - CMS will face similar needs with HGCAL in HL-LHC

- For HL-LHC, computing simulation more crunched
  - Reconstruction usage will scale ~linearly with pileup → less resources for sim.
The Need for Fast Simulation

- Geant4 calo simulation is a significant part of ATLAS computing budget
  - CMS will face similar needs with HGCAL in HL-LHC

Fast & Accurate Calorimeter Simulation is Needed!

- Reconstruction usage will scale \(~\)linearly with pileup
  \(\rightarrow\) less resources for sim.
Diffusion Models

- Gotten very popular for image generation in last few years
  - Recent ‘Stable Diffusion’ (2112.10752) very impressive results
- High quality images, reasonable computation times

“AI aiding physicists at LHC to analyze data and discover new particles”
Starting with some image, **iteratively add Gaussian noise**, eventually reaching pure noise.

Train a model to **invert the noising process**
- Typically train to predict normalized noise component of image, can subtract it off.

Generate by starting from noise image, **iteratively denoise using trained model**.

Can condition on additional input information
- Eg. text prompt or incident particle energy.

NSteps typically $O(1000)$
Dataset: Calo Challenge

- **Community challenge** to compare generative models for Calorimeter simulation
- **Standard datasets to allow comparison**
  - Dataset1: **ATLAS-like** geometry, 5 layer cylinder with **irregular binning**, 368 voxels
  - Dataset2: 45 layers, 6480 total voxels
  - Dataset3: 45 layers, 40,500 total voxels
CaloDiffusion

- We train diffusion models to generate synthetic calorimeter showers based on Geant simulations.
- We use **400 steps** to interpolate from real shower to Gaussian noise.
- Denoising network is has ‘U-net’ architecture based on 3D convolutions
  - Takes as input noisy shower, incident particle energy and step of diffusion process.
- Several novel optimizations utilized.

![Diagram showing the process of CaloDiffusion](image-url)
Optimizing for Cylindrical Data

- Regular convolutions assume pure translation symmetry
- Our data: \( \phi \) is periodic, and \( R \) & \( Z \) not translation invariant

Implement **cylindrical convolutions** to respect periodic boundary of \( \phi \)

Allow convolutions to be **conditional on \( R \) & \( Z \)** by using additional channels

Shower input

\[ \text{‘Radius input’} + \text{‘Layer input’} \]

‘Circularly’ pad \( \phi \) dimension before 3D conv

Additional input channels
Embedding Irregular Geometries

- Dataset 1 (current ATLAS detector) has a cylindrical structure with **irregular binning**
  - Different radial / angular bins in each layer → can’t apply cylindrical convolutions
  - Previous approaches have used fully connect networks or very large 1D CNN’s
- Learn an **embedding** that maps input into **regular cylindrical structure**
Results
Average Showers

**Geant**

Layer 0  | Layer 1  | Layer 2  | Layer 3  | Layer 12
---|---|---|---|---

Energy (GeV)

10^{-3} | 10^{-2} | 10^{-1} | 10^0 | 10^1 | 10^2 | 10^3

**Diffusion**

Layer 0  | Layer 1  | Layer 2  | Layer 3  | Layer 12
---|---|---|---|---

Energy (GeV)

10^{-3} | 10^{-2} | 10^{-1} | 10^0 | 10^1 | 10^2 | 10^3
Dataset 1 Distributions

Energy deposited in layer 0

Energy deposited in layer 2

Energy deposited in layer 12

Voxel energy distribution

Center of Energy in Δφ in layer 2

Center of Energy in Δη in layer 2
Quantifying Performance

- Train a NN classifier to distinguish between Geant showers and CaloDiffusion showers
- Quantify sample quality based on AUC on holdout set

<table>
<thead>
<tr>
<th>Classifier AUC *</th>
<th>Dataset 1 (ATLAS-like)</th>
<th>Dataset 2</th>
<th>Dataset 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>~0.63</td>
<td>~0.6</td>
<td>~0.7</td>
</tr>
</tbody>
</table>

AUC much less than 1 → Very similar showers!

*Preliminary numbers, somewhat dependent on exact classifier training setup
Future Work

- Some “global” properties (ie total shower energy), still need work
  - They generally improve with overall quality, but hard to specifically optimize in diffusion training
  - Could try batch-level MMD loss or similar if needed

- Generation time is somewhat slow compared to other ML approaches because of iterated generation (still faster than Geant)
  - Can be improved with different sampling algos, compression, or distillation methods
  - Or start generation from approximate shower instead of pure noise (“Cold Diffusion”, 2208.09392)

- Extend to more complicated geometries (e.g. CMS HGCal)
• **Diffusion models** are able to generate very high quality showers

• Utilized several optimizations for **cylindrical** calorimeter geometries & new **embedding** approach for irregular shapes

• Classifiers struggle to distinguish between Geant & diffusion showers

• Future work: continue to optimize training, improve generation time, more complicated geometries
● Co-author: Kevin Pedro
● Thank you to the CaloChallenge organizers for the dataset and evaluation code
  - Michele Faucci Giannelli, Gregor Kasieczka, Claudius Krause, Ben Nachman, Dalila Salamani, David Shih and Anna Zaborowska
● Support: Fermi National Accelerator Laboratory, managed and operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy
Technical Details

- ‘logit’ transformation of voxel energies and then standard scale to zero mean and unit variance
  - Correct preprocessing important for diffusion process, related to scale of added noise
- Denoising network uses ‘U-net’ architecture with cylindrical convolutions
  - Two conditional inputs: shower energy and diffusion step
  - ~400k params for dataset 1 and 2, 1.1M for dataset 3
- 400 diffusion steps, ‘cosine’ noise schedule (2102.09672)
- Choices for training objective:
  - Datasets 1 and 2: Network is trained to predict noise component of image
  - Dataset 3: Network trained to predict weighted average of noise component and un-noised image,
    - More stable, recommended by 2206.00364
- Sampling uses DDPM algorithm (2006.11239)
## Additional Metrics

- **Distance metrics:**
  - Frechet Particle Distance and Kernel Particle Distance (proposed in [2211.10295](https://arxiv.org/abs/2211.10295))
    - Use implementation *proposed for CaloChallenge*, based on high level shower features
  - We find that the computation of FPD is slightly biased, ie non-zero values even comparing different random samples of Geant to each other
  - Compare scores for Diffu-Geant (D-G) vs Geant-Geant (G-G)

<table>
<thead>
<tr>
<th></th>
<th>Dataset 1 (ATLAS-like)</th>
<th>Dataset 2</th>
<th>Dataset 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FPD (x10^3)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-G / G-G</td>
<td>35 / 8</td>
<td>107 / 8</td>
<td>275 / 11</td>
</tr>
<tr>
<td><strong>KPD (x10^3)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-G / G-G</td>
<td>6.3 / 0</td>
<td>0.1 / 0</td>
<td>0.7 / 0</td>
</tr>
</tbody>
</table>
Embedding Details

• First find superset of all radial/angular bins → embedding space

• For each layer, embedding in radial dimension is an $M_i \times M_*$ matrix
  
  – $M_i$ ($M_*$) is number of radial bins in layer i (embedding space)
  
  – Initialize weights be proportional to area overlap of bins + $10^{-3}$ * Gaussian noise

• Reverse matrix is $M_* \times M_i$, initialized to pseudo-inverse of embedding matrix

• For now, enforcing phi symmetry, energy is split evenly among phi bins (not learnable)

• Found small benefits of conditioning on phi in addition to R & Z
  
  – There is slight non-uniformity in phi in the energy distributions of dataset1
• Generated calorimeter showers with regular & ‘fast’ version of Geant4

• Use a CNN network to ‘denoise’ fast-sim shower image to match high granularity one

• Decent performance in a relatively simple setup
  – Studies showed adding more info to the network beyond ‘energy image’ only moderately improved performance
  • Tried multiplicity, time of energy deposit, other Geant info
Latent Diffusion Models

- Key advantage is that costly diffusion steps done in smaller latent space
- Relies on encoder not losing any important info
  - ‘perceptual loss’ supposed to reduce blurriness
  - Small regularization of latent space (std. normal KL or vector quantization) during AE training
- Conditioning setup very flexible
  - Text prompts using some language model
  - Image conditioning
  - ...
Stable Diffusion (aka Latent Diffusion)

- First encode your image with an autoencoder to a smaller latent space
  - They used a factor of 4 or 8 for each dimm.
- Transform your conditioning data into a latent rep
- Denoising performed on the latent representation of your image, using conditioned data
  - Conditioning done using an attention mechanism
- Decode back into pixel space
Existing work: CaloScore (2206.11898)

- **Score based** (instead of denoising) diffusion model for calorimeter generation
  - Instead of Gaussian noise, more complicated Markov chain
  - Learn score of the data \( \nabla_x \log(p(x)) \) at each iter \( \rightarrow \) can invert process
  - Results seem \(~\text{ok}\) but computationally expensive \( \rightarrow \) clear room for improvement
  - Converted to cartesian geometry (with some loss of information)

- Some ML literature showing score based and denoising diffusion are connected
  - See eg Appendix B3 of arXiv:2206.00364
Geometry Diagram

Irregular Input → Embed → Cylindrical Input → Denoise → Cylindrical Output → Reverse → Irregular Output